

# Guía de ejercicios # 8: Repeticiones y Máscaras (Q5)

## Organización de computadoras

### UNQ

Los objetivos de esta práctica son:

- Aprender el concepto de iterar y cómo se implementa en la arquitectura Q4
- Entender la motivación para el uso de máscaras y lo necesario para que la arquitectura Q lo implemente.
- Relacionar con conceptos anteriores.

## Repeticiones

1. Escribir **y documentar** una rutina que calcule la multiplicación de los valores que se encuentran en los registros R5 y R6 respectivamente pero sin usar la instrucción MUL.  
Realizar una **prueba de escritorio** con valores que prueben la rutina implementada.
2. Escribir **y documentar** una rutina que calcule la división entera entre los valores de R0 y R1 respectivamente sin usar la instrucción DIV.
3. Escribir **y documentar** una rutina que cuente la cantidad de dígitos "1" en la cadena almacenada en R6.
4. Escribir **y documentar** una rutina que calcule el factorial del valor almacenado en R5. Dicho valor está representado en BSS.

## Operaciones lógicas bit a bit

5. ¿Cuál es el resultado de las siguientes operaciones?

$$(a) \quad \begin{array}{r} \text{AND} \quad 1101 \\ \quad \quad 0111 \\ \hline \quad \quad ? \end{array}$$

$$(b) \quad \begin{array}{r} \text{NOT} \quad 0100 \\ \quad \quad ? \\ \hline \quad \quad ? \end{array}$$

$$\begin{array}{r}
 \text{AND} \quad 1010 \\
 \quad \quad 1100 \\
 \hline
 \quad \quad ? \\
 \text{(c) OR} \quad 0101 \\
 \quad \quad ? \\
 \text{XOR} \quad 1100 \\
 \hline
 \quad \quad ?
 \end{array}$$

6. Complete con el operador adecuado (AND, OR, XOR, NOT) en las operaciones. Ejemplo:

$$1001 \dots 1001 = 0000 \rightarrow \mathbf{1001 \text{ XOR } 1001} = 0000$$

- (a)  $1000 \dots 1011 = 1011$   
 (b)  $1111 \dots 0011 = 0011$   
 (c)  $\dots 0011 = 1100$

7. Complete las operaciones logicas dada una cadena de bits formada por  $(x_7x_6x_5x_4x_3x_2x_1x_0)$ . Ejemplo:

$$\begin{array}{r}
 \text{OR} \quad x_7x_6x_5x_4x_3x_2x_1x_0 \\
 \quad 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \\
 \hline
 \quad 1 \ x_6 \ 1 \ x_4 \ 1 \ x_2 \ 1 \ x_0
 \end{array}$$

$$\begin{array}{r}
 \text{(a) OR} \quad x_7x_6x_5x_4x_3x_2x_1x_0 \\
 \quad 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \\
 \hline
 \quad ?
 \end{array}$$

$$\begin{array}{r}
 \text{(b) AND} \quad x_7x_6x_5x_4x_3x_2x_1x_0 \\
 \quad 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \\
 \hline
 \quad ?
 \end{array}$$

$$\begin{array}{r}
 \text{AND} \quad x_7x_6x_5x_4x_3x_2x_1x_0 \\
 \quad 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \\
 \hline
 \quad ? \\
 \text{(c) OR} \quad 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \\
 \quad \quad ? \\
 \text{XOR} \quad 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \\
 \hline
 \quad ?
 \end{array}$$

## Programas en Q5

8. Implementar la rutina `absolute` siguiendo su documentación:

absolute	
<b>Requiere</b>	Una cadena en CA2(16) en el registro R0
<b>Modifica</b>	COMPLETAR
<b>Retorna</b>	El valor absoluto en R0, donde el primer bit se cambió por un 0

9. Implementar la rutina `xor` en función de su documentación:

xor	
<b>Requiere</b>	Dos valores de 16 bits en R6 y R7
<b>Modifica</b>	COMPLETAR
<b>Retorna</b>	En R7 el OR Exclusivo (bit a bit) entre R6 y R7

10. Implementar la rutina `invImp` que dada una cadena de 16 bits en R1, invierta el valor de las **posiciones impares**. Utilice máscaras y operaciones lógicas.
11. Implementar la rutina `opuesto`, que calcule el opuesto aditivo del número (módulo) almacenado en el registro R2 sin usar SUB. Dicho número está representado en  $CA2(16)$ .
12. Un registro meteorológico es aquel donde sus bits indican la precipitación en cada día, para una estación meteorológica determinada, durante 14 días (los 14 bits de la derecha). En el sistema occidental se indica con 0 si llovió y con un 1 el caso contrario. En el sistema oriental se indica al inverso. Por ejemplo, la cadena 0011001100110011 se convierte así: 0000110011001100 (notar que los primeros 2 bits siempre están en cero).

Implementar la rutina `occidentalAOriental` según la documentación:

occidentalAOriental	
<b>Requiere</b>	En R3 un registro meteorológico en sistema OCCIDENTAL
<b>Modifica</b>	COMPLETAR
<b>Retorna</b>	El registro R3 convertido al sistema ORIENTAL

13. Implementar la rutina `diasDeLluvia` según la documentación:

diasDeLluvia	
<b>Requiere</b>	Un registro meteorológico en sistema oriental en R6
<b>Modifica</b>	COMPLETAR
<b>Retorna</b>	En R2 La cantidad de días de lluvia registrados

**Pista:** puede usar la rutina `desplazarIzq` que tiene la siguiente documentación:

desplazarIzq	
<b>Requiere</b>	Una cadena en R0
<b>Modifica</b>	—
<b>Retorna</b>	En R1 desplaza los bits de la cadena contenida en R0 un lugar hacia la izquierda

14. La siguiente es la codificación de permisos de acceso sobre archivos en un sistema Linux:
- Con 3 bits se indica:
    - (a) Permiso lectura (r)
    - (b) Permiso escritura (w)
    - (c) Permiso ejecución (x)

- Con 3 cadenas de 3 bits cada una, se describen permisos de usuario, grupo y otros

La cadena resultante respeta el siguiente formato: **rwxrwxrwx**, donde cada subcadena de 3 bits corresponde al usuario, al grupo y a otros respectivamente. Por ejemplo, la cadena **111111111** le da todos los permisos a todos.

Implementar la rutina **groupWriting** que, dada una cadena que codifica los permisos de acceso a determinado archivo, determine si otro usuario del grupo lo puede escribir. Dicha cadena esta almacenada en R4, y el programa debe poner un 1 en R5 si es posible, y 0 en caso contrario.